

# Algorithm development based on artificial intelligence for function manipulation and comparison with numerical techniques used in engineering teaching

Caio Luiz Nunes<sup>1</sup>, Elias Rocha Gonçalves Júnior<sup>2</sup>, Virgínia Siqueira Gonçalves<sup>1</sup>, Bianca Siqueira Gonçalves<sup>1</sup>, Elias Rocha Gonçalves<sup>1</sup>

<sup>1</sup>CEPECAM, Candido Mendes University, Brazil

<sup>2</sup>Engineering Coordination, Multivix College, Brazil

**Abstract**— This paper aims to develop and verify the efficiency of algorithm applications and compare them with the numerical techniques used in engineering areas, stem from structures of neural network for function approximation. For that, function modeling were used, based on complex formulas, making approximation mandatory. The statement investigated is that neural networks are capable to solve with great accuracy problems of this kind. As a test model, a measurement approximation of thermodynamics tables (saturated water) was carried and it was compared with the results of traditional methods (linear and quadratic functions), confirming the efficiency and possibility of its use.

**Keywords**— *neural networks, artificial intelligence, engineering education.*

## I. INTRODUCTION

Artificial Intelligence (AI) consists in withdraw information from the embbded ambient and utilize them to solve determinated problems, like a study. This way, it becomes adaptive for use in a variety of areas of knowledge, as engineering [1]. According to Heckmann [2], Artificial Intelligence is a Science to make a computer do tasks that, until now, only people could do it and very capacited people could do well.

The engineering is visible everywhere, their working fields are extremely embracing, making possible to preside in small domestic tubulation system and even in great petrolific industry. With the technology evolution, the engineer obtained more efficient and faster ways to mathematically calculate using the computer. It can reach even higher levels solving math problems. Therefore, it is necessary to have engineers capable to use the resources of computer Science, since the integration of these fields can be very effective in problem solving.

The AI show itself as an efficient alternative to be studied. Among its capacities are problems involving proximity and function modeling, which can be applied areas such as thermodynamic and function optimizer

problem solving with lots of variables and common restriction in project area [2]. According to Russel and Norvig [3], learn how to approximate function is considered inductive task, the Artificial Neural Networks (ANN) learns to represent function effectively and usefully.

In this research, it will be compared the function approximate method obtained by artificial neural network and by numerical analysis method, such as interpolation and spline functions, that can be defined by polinomes [4]. As an object, the thermodynamics tables will be used to find a function that approximate the non-registered results to solve specific problems.

At the same time, the constants involved in the estimates of the approximation error for neural networks often depend exponentially or polynomially on the dimensionality of the data [5,6]. Thus, by using lower dimensional data following the mapping of the original data into the lower dimensional space, it can be expected that the error of the approximation will be reduced [7].

This paper aims to develop algorithm that can test Artificial Intelligence techniques (neural network) and compare its practical efficiency with numerical techniques

(interpolations), to verify a possibility of its use in solving common problems on Mechanical Engineering. Simple experiments will be made using algorithms, which will be possible to calculate its efficiency regarding other methods.

## II. LITERATURE REVIEW

Function approximation is a core task in many engineering, economic, and computational problems [8]. There are many approaches to the function approximation including relatively simple methods such as least squares linear approximation and many more complex methods such as approximation with splines or neural networks [9].

Functional neural models can also be created by translating well understood principles from classical algorithms from the fields of machine learning [10]. Such techniques work as a data-driven brain models and the hardware and can be used for creating entire cognitive architectures with the potential of helping to decipher neural principles [11].

### 2.1 Linear Interpolation

Linear Interpolation is the simpler case and also the most used. Given two distinct points in a function, a degree 1 polynomial must be created  $P_1(x) = a_0 + a_1x$ , which has to satisfy the equation system, where geometrically  $P_1(x)$  is an equation of a line that pass through two points,  $x_0$  and  $x_1$  [12].

The interpolation function always pass by known points of the original function and, from them, are estimated the unknown values. Being  $f(x)$  the original function and  $g(x)$  the approximation, it can be realized that the distance between them rises when they are more distant of the points used on interpolation. This distance is that defines the error [13].

In practical aspects, functions with more than six points are rarely used, because the error overly rises, so, surpassing to splines in the other cases. The splines are used to get a reduction of the approximate function. The utilized method for his approximation consists in adjusts the polynomial of lower degree than the data subset. These polynomial links are called spline functions [13].

Higher degree polynomials tend to not to perceive abrupt changes in the function behavior. Polinômios com ordem alta tendem a não captar mudanças bruscas no comportamento da função. Thus, the splines obtain better approximation in functions with this kind of characteristic [14].

### 2.2 Neural Networks: learning and implementation

As Braga et al. [15] stated, the neural network training can be supervised or unsupervised. While the unsupervised training doesn't require a desired output (the network makes an auto organized considering only the input data), the supervised training considers the network learning from input data and the respective desired outputs.

While, for a satisfactory performance of a RNA is necessary to properly choose the network attributes. There are lots of learning algorithms developed for RNA training, and also needed to define the number of network layers, the neurons of hidden layers, the activation function of hidden layers and output and the specifics parameters that compose each algorithm [16].

The main characteristic of neural network is its adaptation and generation capacity. This feature is acquired changing its synaptical weights in such a way that minimizes the network output error, until an optimum solution is reached. To achieve such solution, some specific training algorithms from each neural network architecture are used.

The Backpropagation algorithm was developed by Werbos [17] and by Parker [18]. Since its creation, the backpropagation algorithms has been largely used as a learning algorithm for RNA with multiple layers' topology [19-21]. For this paper, Multilayer Perceptron (MLP) network was used, which is based in this algorithms (or variations) for its training [2].

### 2.3 Thermodynamic Tables

The comprehension of these tables are very important for academical develop of a mechanical engineer, because they are indispensable when working with steam or thermodynamic cycles. Temperature and pressure can be expressed in three occasions: compressed liquid, superheated steam and saturated steam [22].

The tables have rich content of details, low error tax from real of compounds and lots of temperature versus pressure are already listed, besides they are constantly used with linear interpolation by engineer students. With these characteristics, the tables are a good option to be used in the approximation method, because, besides showing practical values, also shows simplicity in its use [22].

## III. METHODOLOGY

Through testing of the studied methods of this paper, we developed algorithms wrote in C language, and Matlab was used for graph generation, with that it is possible to obtain better view and results analysis.

Every code used in this research is owned by the authors and are available on the link “<https://github.com/Eliasrgjunior1/CITI2017-1/blob/master/README.md>”, any adaptation or parts from others authors were used. Its development was based on techniques described during the literature review, highlighting the AI fundamentals.

About function approximation case, thermodynamics table (saturated steam water), which calculations using interpolation (linear, quadratic and spline function) were made and neural networks were used as object of study. A routine was generated, using the Matlab, so the graphs for every function using each method can be plotted. To model the functions, 46 samples were selected from the table randomly for Neural Network training and interpolation. The efficiency calculation for each technique was done using the thirty-three unused samples. Comparing the values obtained with the real ones, it was possible to estimate the error in both methods, showing the most efficient one in this context.

For each case with neural network were used, its structure was made in Multilayer Perceptron's (MLP) architecture and its training was the Levenberg Maquardt (LM), so the used topologies can be varied to obtain results more similar with the real ones.

Through these methods, this research aims to obtain desired solutions about the possibilities of application of neural network on engineering, its efficiency and the facility of the projects.

### 3.1 Data preparation

In order to the data be suitable to test realization, the following steps were made:

- Creation of a file containing the values to be used from thermodynamic tables of saturated water, focusing on values of pressure and enthalpy;
- Development of the routine for the Matlab plot graphs extracted from the developed algorithms. These ones will be plotted in two dimensions with pressure to temperature;
- Tabled point plotting to visualize the table clearly.

### 3.2 Algorithm Programming

In this phase, the following actions were done:

- Development of an artificial neural algorithm (ANA) through MatLab®, with a hidden layer, and LM training. The algorithm will be capable to alter the number of neurons of the hidden layer, to do the tests with many topologies;
- Development of an algorithm to linear interpolation using splines;

- Development of an algorithm for interpolation, using quadratic spline functions, that must receive the number of points and the needed table points to approximate the function.

### 3.3 Tests

The tests followed the process defined by the flowchart (Fig. 1):

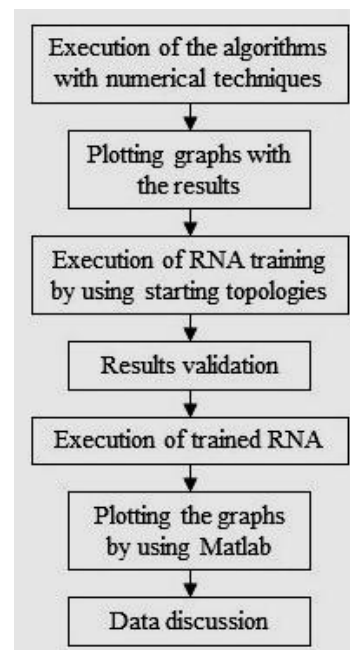


Fig. 1: Process flowchart.

## IV. RESULTS AND DISCUSSION

### 4.1 Saturated Water Thermodynamic Table

The graphs of thermodynamic tables obey a determined pattern, which has to be learnt by the Neural Networks (NN). For validation, the points of the table were placed in order, generating the graph presented in Fig. 2).

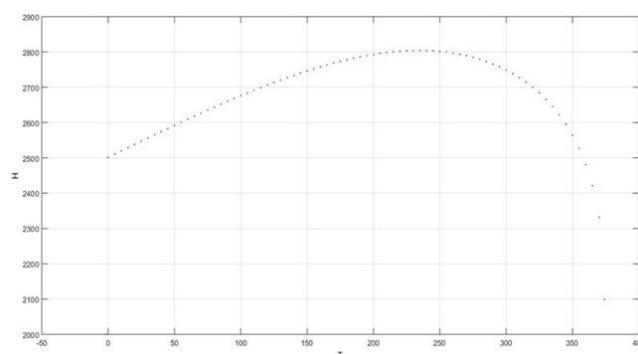


Fig. 2: Temperature (°C) x enthalpy (kJ/kg) graph.

#### 4.2 Dispersion Calculation

The algorithm used for interpolation worked like as expected, modeling the function with low error rate. From trained neural network and interpolation, desired values approximations were obtained. As result, the variance and standard deviation were calculated (Table 1).

Table.1: Results of the calculation of standard deviation and variance – temperature x enthalpy.

Tests	Standard Deviation	Variance
Linear Interpolation	0,35438	0,125585
Quadratic Interpolation	0,017839	0,000318
NN1	0,829772	0,688521
NN2	0,004884	0,000238
NN3	0,001444	0,002085
NN4	2,231818	4,98101
NN5	1,2319894	1,520028
NN6	4,299726	18,487644

#### 4.3 Approximation graph analysis

The algorithms used in interpolation worked as expected, modeling the function with low error rate. From trained neural networks and interpolations, the approximations for desired values were obtained.

Graphs comparisons between the NN of 1 or 2 layers and interpolations were made and better efficiency obtained can be noticed. Fig. 3 and Fig. 4 shows the graphs of combined NNs and points of interest. Fig. 5 demonstrates the values obtained by polynomial interpolations. To ensure a high-quality product, diagrams and lettering must be either computer-drafted or drawn using India ink.

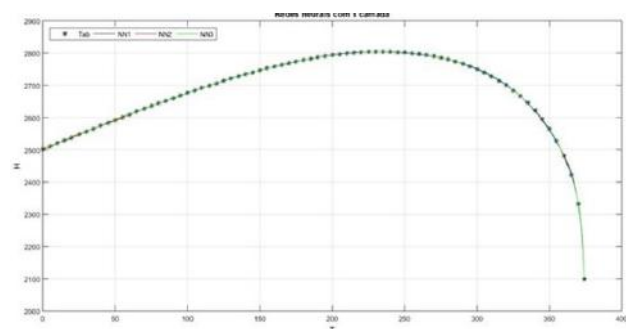


Fig. 3: Temperature (°C) x enthalpy (kJ/kg) graph. - NN of 1 hidden layer approximation.

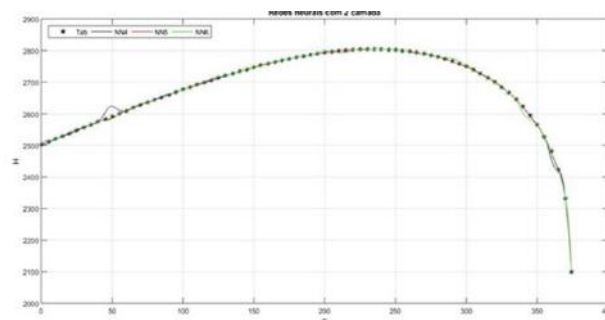


Fig. 4: Temperatura (°C) x enthalpy (kJ/kg) - NN of 2 hidden layers approximation.

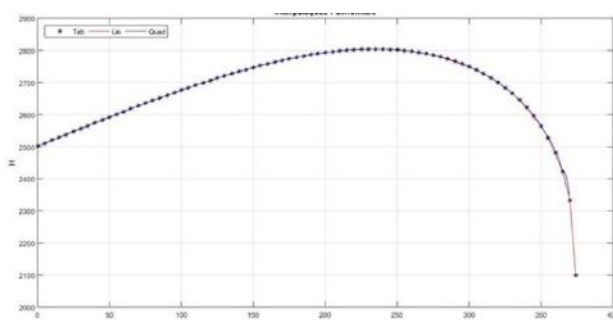


Fig. 5: Temperature (°C) x enthalpy (kJ/kg) - approximation with polynomial interpolation.

At last, for comparative effects, a combined graph of the neural network and polynomial interpolation values were plotted as presented in Fig. 6.

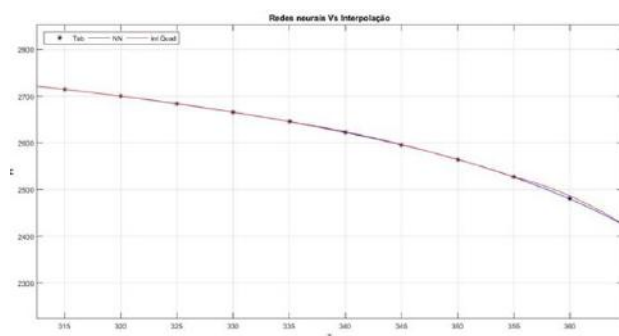


Fig. 6: Temperature (°C) x enthalpy (kJ/kg) comparing the obtained values by the approximation techniques used in this study.

The graphs show the approximation results, using all of the points. The neural networks didn't behave in an exact way, it may contain errors, including the points that were used for training. It is important to state that this behavior doesn't occur in interpolation, because with a proper training can eliminate this kind of error. Also, there is a homogeneous behavior in the whole function, where the



error is not increased, comparing with the interpolation, they have more efficient approximation in these points.

It's possible to notice, by the graphs obtained by the used table, that it generates a continuous function with insignificant randomness. This is the best scenario for linear and quadratic interpolation to be applied. In situations that high precision is not necessary, the linear interpolation would be ideal, since it's simpler, but, when the application requires high precision, the methods must be compared.

In every test, the neural network configuration number 2 could stand out from the others, so this will be the pattern for comparison purposes.

It also can be noticed that the linear interpolation falls short when the curves with higher inclination, making the error, in some cases, rise considerably. The obtained curve by quadratic interpolation wasn't symmetrically arranged, which deviates the function from satisfactory results.

The NN must have its training meticulously made to have an efficient result. Some configurations show an error rate higher than the others, but the final result obtained was more precise than quadratic interpolation. The NN's dispersions were lower or equal in all situations, as displayed in the tables.

From the training, it can be observed that the needed error rate for this kind of function is from  $10e-2$  order, lower errors rate configures about the training, which is the two-layered neural network's case. Besides, it is possible to realize that, with less complex functions, simpler NN, with a layer and few neurons, are more efficient for this objective. The training with few samples (relatively to NN) was possible and the neural network adapted themselves without great effort.

## V. CONCLUSION

Based on the superior performance met by neural network on interpolations, even in standard functions with low randomness, it's assured that they can be used to complete the measurement table (like thermodynamic ones), adapt the control answer description function, vectorial drawing in specialized software, like Matlab and help in physics phenomenon's study. Besides thermodynamical for function description, and even help to solve and develop specifics engineer.

The most efficient NN's topologies – NN2 and NN3 – can be used by starting point to similar projects, the described training can be utilized in other problems. The Neural Network can also be used in cases that have function mathematical solution, like this one, and presents

similar results or more efficient ones, since the adequate training routine is followed.

Based on the results reported, some other comparison of numerical analysis can be explored, like extrapolation and some function modeling that can be difficult to calculate, also be able to compare between the others variables provided on thermodynamical tables.

## REFERENCES

- [1] Silva, I. N., Spatti, D. H., & Flauzino, R. A. (2010). Redes neurais artificiais para engenharia e ciências aplicadas. São Paulo: Artliber Editora.
- [2] Heckmann, J. R. (1999). Identificação de Potenciais Aplicações da Inteligência Artificial na Metrologia. Master dissertation, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.
- [3] Russel, S., & Norvig, P. (2013). Inteligência Artificial (3. ed.). São Paulo: Campus.
- [4] Paulino, C. D., Pestana, D., Branco, J., Singer, J., Barroso, L., & Bussab, W. (2011). Glossário Inglês-Português de Estatística (2. ed.). Sociedade Portuguesa de Estatística e Associação Brasileira de Estatística.
- [5] Hornik, K., Stinchcombe, M., White, H., & Auer, P. (1994). Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Computation*, 6(6), 1262-1275.
- [6] Kainen, P. C., Kurkova, V., & Sanguinetti, M. (2012). Dependence of computational models on input dimension: Tractability of approximation and optimization tasks. *IEEE Transactions on Information Theory*, 58(2), 1203-1214.
- [7] Andras, P. (2014). Function Approximation Using Combined Unsupervised and Supervised Learning. *IEEE Transactions on Neural Networks And Learning Systems*, 25(3), 495-505.
- [8] Haykin, S. (2008). *Neural Networks and Learning Machines*. Englewood Cliffs: Prentice-Hall.
- [9] Anastassiou, G. (2011). Multivariate sigmoidal neural network approximation. *Neural Networks*, 24(4), 378-386.
- [10] Serrano-Gotarredona, T., Linares-Barranco, B., Galluppi, F., Plana, L. A., & Furber, S. (2015, May). ConvNets Experiments on SpiNNaker. *Proceedings of the IEEE International Symposium on Circuits and Systems*, Lisbon, Portugal, 2405-2408.
- [11] Eliasmith, C., Stewart, T., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A Large-Scale Model of the Functioning Brain. *Science*, 338(6111), 1202-1205.
- [12] Nepomuceno, L., Baptista, E. C., Balbo, A. R., & Soler, E. M. (2015). Coevolutionary Genetic Algorithm Based on the Augmented Lagrangian Function for Solving the Economic Dispatch Problem. *IEEE Latin America Transactions*, 13(10), 3277-3286.
- [13] Ruggiero, M. A. R., & Lopes, V. L. R. (1996). *Cálculo numérico: aspectos teóricos e computacionais* (2. ed.). São Paulo: Pearson Makron Books.

- [14] Pilling, S. (2009). Cálculo Numérico [Apostila da Faculdade de Engenharia, Arquiteturas e Urbanismo] São José dos Campos: UNIVAP.
- [15] Braga, A. P., Carvalho, A. P. L. F., & Ludermir, T. B. (2007). Redes neurais artificiais: teoria e aplicações (2a ed.). Rio de Janeiro: LTC Editora.
- [16] Haykin, S. (2001). Redes neurais: princípios e prática (2. ed.). Porto Alegre: Bookman.
- [17] Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Doctoral dissertation, Harvard University, USA.
- [18] Parker, D. (1982). Learning Logic (Intention Report/1982). Stanford, UK, Office of Technology Licensing, Stanford University.
- [19] Nascimento, C. F., Oliveira, A. A., Goedel, A., Silva, I. N., & Serni, P. J. A. (2010). Neural Network-Based Approach for Identification of the Harmonic Content of a Nonlinear Load in a Single-Phase System. *IEEE Latin America Transactions*, 8(1), 65-73.
- [20] Nascimento, M., Peternelli, L. A., Cruz, C. D., Nascimento, A. C. C., Ferreira, R. P., Bhering, L. L., & Salgado, C. C. (2013). Artificial neural networks for adaptability and stability evaluation in alfalfa genotypes. *Crop Breeding and Applied Biotechnology*, 13, 152-156.
- [21] Melo Junior, J. G., Farias, F. S., & Kato, O. R. (2016). Assessment of the Sustainability of Agroecosystems in the Amazon Region Using Neural Artificial Networks. *IEEE Latin America Transactions*, 14(8), 3804-2810.
- [22] Borgnakke, G., & Sonntag, R. E. (2013). Fundamentos da Termodinâmica (8a ed.). São Paulo: Blucher.